Vol.2021-CG-184 No.16
Vol.2021-DCC-29 No.16
Vol.2021-CVIM-227 No.16
2021/11/5

画素対応を用いた自動着色

沖川翔太[1]　森島繁生[2]

概要：アニメ制作過程において，線画に対して色を塗る「彩色」の工程には膨大な負担がかかっている．そのため，「彩色」の工程の負担軽減のために自動化の需要が存在する．本研究では複数の参照画を用意し，対象の線画に対して自動で着色を行うことを目的とする．まず複数の参照画の線画と，着色対象の線画の画素ごとの対応を取得する．そして，対応関係を利用して画素ごとに参照画から色を取得することで線画の着色を行う．各画素について複数の参照画のうち，類似度の最も高い参照画から色を取得する．実験を行った結果，従来手法よりも精度の高い着色を行えることが示された．

キーワード：画像処理，画素対応，自動着色

1. はじめに

アニメ制作は「企画・脚本・原案」「キャラクターデザイン・イメージボード」「絵コンテ・作画・彩色」「撮影・編集」などといった様々な工程が存在する．「企画・脚本・原案」においてはアニメをどのように制作するかを考え脚本家が台本を書き，それらを元にして絵コンテを作成する．「キャラクターデザイン・イメージボード」においては登場人物のデザインや物語の絵を作成する．「絵コンテ・作画・彩色」においては原画マンが絵コンテを基にして原画を作成し，動画マンが清書して線画を作成する．その後彩色担当のスタッフが一枚一枚色を塗っていく．「撮影・編集」においては完成した作画を基に編集作業で特殊効果などを加えながら映像を作成する．これらの作業のうち，「彩色」の工程を自動化する恩恵はとても大きいと考えられる．なぜなら彩色という工程は色を塗るという単調な作業であり，また膨大な負担がかかる工程であるからである．一本のアニメーションは 24 フレーム×60 秒×30 分≒約 4 万フレームで構成されており，これらを全て彩色するためには手間と時間の点で膨大なコストがかかってしまう．そのため，本研究では彩色を自動化することで彩色にかかるコストを軽減することを目的とする．

線画の自動着色を行う方法として，深層学習を用いたものが多数存在する．深層学習を用いたものは参照画から線画の着色を行うものと，参照画を必要とせず線画の着色を行うものが存在する．参照画から線画の着色を行うものにはイラストレータ用の自動着色を目的としたものが存在する．これらはグラデーションなどの効果を含めたイラストの見栄えに重点を置いたものであり，領域ごとに単色で塗るアニメ制作には対応していない．また，アニメ制作に対応している自動着色手法は，1 枚のみを参照画としているために遮蔽などに弱いという欠点や，新しいキャラクターに対して転移学習をする必要があるなどの欠点が存在する．参照画を必要としない自動着色手法では事前に大量の画像を必要とするため，新しい作品やキャラクターに対して汎用性が低いという欠点が存在する．

本研究では着色対象画像の線画と複数の参照画の線画との画素の対応関係を取得し，その対応関係を利用して着色対象の色を塗っていく．

2. 関連研究

2.1 線画の自動彩色

線画の自動彩色の手法は深層学習を用いる手法とグラフマッチングを用いる手法などが存在する．

深層学習を用いる手法としては，まずユーザーがヒントを与えることで自動着色をするものが存在する．これには，ドラフト工程とリファインメント工程という二つの工程を行うことで色鮮やかに仕上げる手法[1]や，ユーザーのヒントをクラスタリングしてスケルトンマップを用いてアニメのような仕上がりの着色を行う手法[2]が該当する．これらの手法はユーザーがヒントを与える必要があり，アニメ制作の時間的負担の軽減にはなっていないと考えられる．次に，参照画から線画の自動着色を行う手法が存在する．これ

1 早稲田大学
Waseda University
2 早稲田大学理工学術院研究所
Waseda Reserch Institute for Science and Engineering

Vol.2021-CG-184 No.16
Vol.2021-DCC-29 No.16
Vol.2021-CVIM-227 No.16
2021/11/5

には Color Transform Network と Temporal Constraint Network を用いることで時系列的に滑らかでかつ整った着色を行う手法[3]や参照画の線画と対象画像の線画の対応関係を取る CMFT モデルを提案した手法[4]，パッチベース学習を用いて自動着色を行う手法[5]が該当する．1 つ目の手法は新しいスタイルのアニメに対応するためにはその都度転移学習を行う必要があり，2 つ目の手法は 1 枚のみを参照画とするため遮蔽などに弱いという欠点が存在する．3 つ目の手法はシーンごとに学習が必要であるという弱点が存在する．次に，ユーザーのヒントや参照画像を用いずに着色を行う手法が存在する．これには cGAN を用いる手法[6]や，Unet を用いて各ピクセルがどのクラスに分類されるかを予測することによって着色を行う手法[7]が該当する．後者の手法においては信頼度マップを作成することによって，誤った予測をした可能性のある箇所を示すことも提案している．これらの手法は学習に用いたキャラクターの着色にのみ対応しており，学習に用いていないキャラクターの着色には対応していないため汎用性が低いという欠点が存在する．

グラフマッチングを用いる手法として，まず色の領域の関係をグラフとして表す手法[8]が挙げられる．この手法においては，色の領域の関係をグラフとして表した後にグラフマッチングを用いることで参照画像と対象の線画で領域を対応させて色を塗っていく．また，参照画と対象の線画のグラフの類似度を定式化することで，この手法を参照画が複数となっても可能となるように拡張した手法[9]も存在する．これらの手法においては参照画と対象の線画においてキャラクターの体勢が大きく変化するなど，参照画と対象の線画と領域の数が大きく異なっていると着色が難しいという欠点が存在する．

また，白黒画像に対して自動着色を行う手法[10]も存在する．この手法においては，参照画像と着色対象の白黒画像に対してそれぞれ特徴点を取得した後，特徴点の対応づけを行い着色を行う．この手法においては，大域的な情報を用いて対応付けを行っていないという欠点が存在する．

2.2 画素対応

画像間における対応をとる手法としては SIFT Flow[11]や PatchMatch[12]などが存在する．SIFT Flow では Scale-Invariant Feature Transform (SIFT)[13]特徴量を用いて画像間の密な対応を取得する．この方法においては SIFT 特徴量を用いているため照明変化や回転，拡大縮小に強いという特徴がある．この手法は色の勾配に着目しているが，線画においては色の勾配が少ないため，線画における記述子として SIFT 特徴量は適切でない．PatchMatch では，画素間の隣接関係を利用してパッチを用いた画像間の最近傍探索を高速に行うことができる．しかし線画のような色の勾配が少ない画像においては，大域的な対応を取得することは困難である．

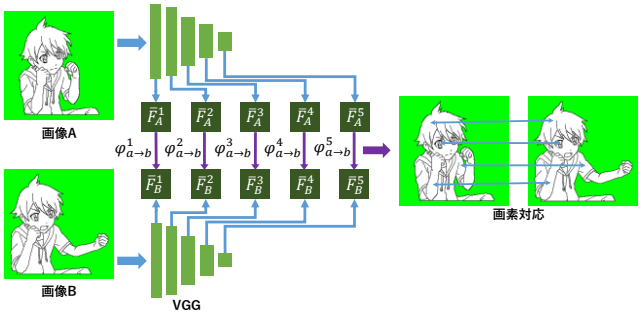

図 1　画素対応

深層学習を用いて 2 つの画像間の画素対応を取る手法としては Deep Image Analogy[14]や Neural Best-Buddies[15]が挙げられる．Deep Image Analogy においては VGG19[16]の中間特徴量に対して PatchMatch を適用することによって，画像間の大域的かつ密な画素対応を取得することを目的としている．Neural Best-Buddies は Deep Image Analogy の手法を基にしており，PatchMatch の際に検索対象領域同士の平均と分散を近づける Instance Normalization を行うことによって，2 つの画像に映っている物体が似ていなくても画像間の疎な対応を取得することが出来る．今回は線画同士の密な対応を取得する必要があるため，画像間の画素対応を取得する手法として Neural Best-Buddies ではなく，Deep Image Analogy を採用した．

画素対応を応用したものとしては，アニメの異常検知[17]や顔のテクスチャー合成[18]が挙げられる．アニメの異常検知においては，ターゲットフレームとその前後フレームとの画素対応を取ることで色の塗り間違いを検知している．顔のテクスチャー合成においては，複数の画像から画素対応を取ることによって顔の合成を行っている．

3. 提案手法

3.1 画像間の画素対応の取得

図 1 に画素対応の取得の概略を示す．本手法においては，画像間の画素ごとの対応関係を用いて自動彩色を行う．画像間の対応関係を取る方法としては Liao らの手法 [14]を用いる．ここでは，VGG19 の中間特徴量に対して PatchMatch を用いた最近傍探索を行うことで二つの画像A, Bの対応する画素を決定する．VGG19 の中間特徴量に対する最近傍探索は次の式に基づいて行われる．

$$\varphi^{l}_{A\to B}(p) = \underset{q}{\operatorname{argmin}} \sum_{x\in N(p), y\in N(q)} \left(\left\| \bar{F}^{l}_{A}(x) - \bar{F}^{l}_{B'}(y) \right\|^2 + \left\| \bar{F}^{l}_{A'}(x) - \bar{F}^{l}_{B}(y) \right\|^2 \right) \quad (3.1)$$

$\varphi^{l}_{A\to B}$は VGG19 の relu_1層$(l = 1 \dots 5)$の出力における画像Aから画像Bへの対応関係を表す．$N(p)$は座標p周辺の特徴量パッチを表す．$\bar{F}^{l}_{A}$，$\bar{F}^{l}_{A'}$，$\bar{F}^{l}_{B}$，$\bar{F}^{l}_{B'}$はそれぞれ画像A，画像A'，画像B，画像B'の relul_1 層の出力を表している．ただし，画像A'は画像Aが画像Bの対応する画素値を参照して得られる画像，同様に画像B'は画像Bが画像Aを参照して得られる画像を表す．式(3.1)の右辺で画像Aを VGG19 に入力し

Vol.2021-CG-184 No.16
Vol.2021-DCC-29 No.16
Vol.2021-CVIM-227 No.16
2021/11/5

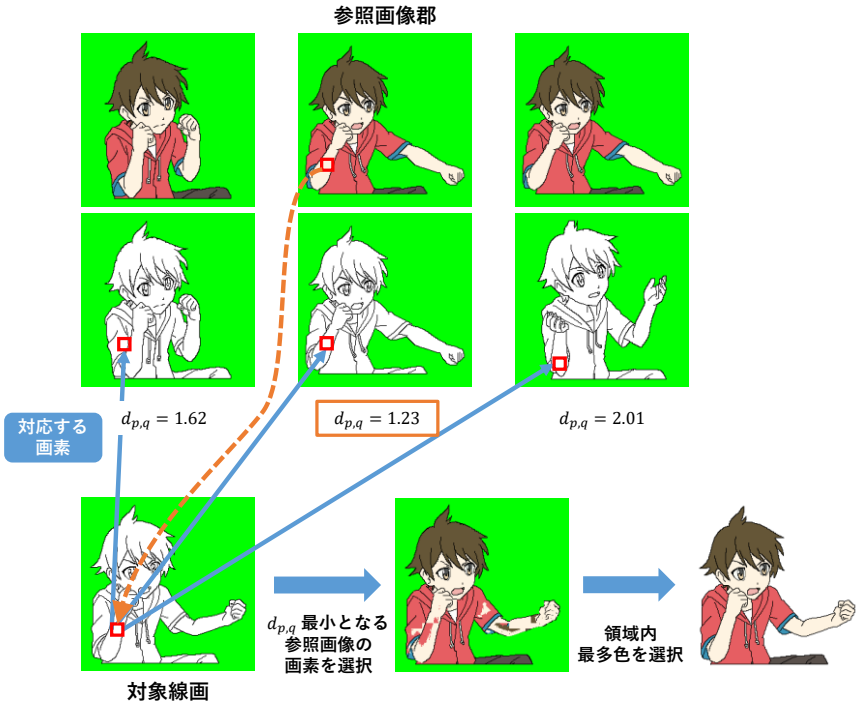

図 2　提案手法の概略図

たときの relu_1層の点pと画像Bを VGG19 に入力したときの relu_1層中の点とのパッチの距離を定義し，その中で最近傍の点を点pに対応する点(点q)としている．

3.2 自動着色

図 2 に提案手法の概略を示す．人間が参照画を用いて線画に着色を行う際，線画の色を塗りたい場所が参照画のどこに対応しているかを探索し，対応している部分から色を取得することで着色を行っている．これを模した操作を行う．

N枚の参照画を$R_i(i = 1,2,\cdots,N)$，参照画の線画をRL_i，着色対象の線画をTLとする．

着色対象の線画TLの点pに対して参照画の線画をRL_iの点q_iが対応しているとする．ここで，以下の量d^l_{p,q_i}を定義する．

$$d^l_{p,q_i} = \sum_{x\in N(p),y\in N(q_i)} \left(\left\|\bar{F}^l_{TL}(x) - \bar{F}^l_{RL_i'}(y)\right\|^2 + \left\|\bar{F}^l_{TL'}(x) - \bar{F}^l_{RL_i}(y)\right\|^2\right) \quad (3.2)$$

VGG19 の中間特徴量は画像の意味的な情報を表している．この事と 3．1 章で議論されていることから，d^l_{p,q_i}はrelu_1層の出力から点pと点q_iの意味的な距離を表していると考えられる．点pの色を決定する際には，点q_iの中で最も点pに意味的に近い点，つまりd^l_{p,q_i}の値が最も小さい点から色を取得すればよい．このことは，以下のように表される．

$$TL(p) = R_i(q_i)\ \ s.t.\quad i = \underset{j}{\operatorname{argmin}}\, d^l_{p,q_i} \quad (3.3)$$

$TL(p)$や$R_i(q_i)$はそれぞれ着色対象の線画TLの点pにおける画素値，参照画R_iの点q_iにおける画素を表す．

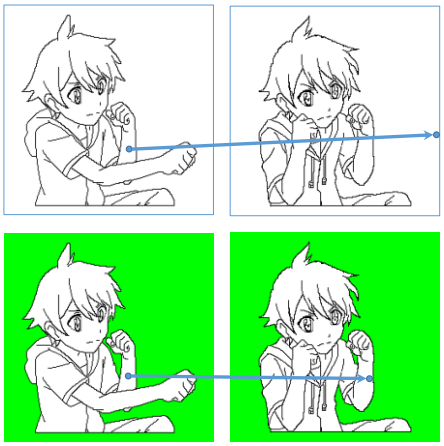

図 3　前処理による対応関係の影響

VGG19 の深い層の出力ほど，より意味的な情報を含んでいる．今回はより意味的な距離によって取得する色を決定したいため，$l = 5$で提案手法を行う．

3.3 前処理

前処理なしに線画同士で対応を取得しようとすると，設定画においてキャラクターの外側の部分(色がない場所)と対応付けてしまう事があり，着色に悪影響が出ることが多々あった．そこで線画においてキャラクターの外側を緑色で塗るというマスク処理を行ってキャラクターではない場所を明示すると，キャラクターの外側部分と対応付けることが少なくなった．そこで，線画においてキャラクターの外側を緑色で塗るという動作を前処理として行った．これを図 3 に示す．

Vol.2021-CG-184 No.16
Vol.2021-DCC-29 No.16
Vol.2021-CVIM-227 No.16
2021/11/5

図 4　自動彩色の結果の例　一番上の行の 3 枚が参照画．それ以外の行は左から a)入力線画，b)Maejima らの手法による彩色結果，c)提案手法を VGG19 の重みで行った彩色結果，d)提案手法を Illustaration2Vec の重みで行った結果，e)正解画像である．Maejima らの手法による彩色結果において緑色になっている箇所は領域の対応付けがうまくいかなかったことを意味する．

3.4 後処理

アニメ制作で制作される画像は，一つの領域が一つの色で塗られている．そのため，本手法の後処理として，領域ごとに最も占める画素の割合が多い画素値でその領域を塗りつぶすという処理を行った．線画の領域を割り出す方法としては trapped ball segmentation[19]を用いた．

4. 実験

4.1 データセット

実験は参照画 3 枚，着色対象の線画 7 枚で行った．中間特徴量に対して PatchMatch を適用する際のパッチサイズは $L = 5$から順に$[9, 9, 9, 11, 11]$である．を参照画と線画はともにサイズは 512×512 で統一した. アニメ制作過程の「彩色」の段階で着色を行うことを考慮して，エフェクトの影響をなくすために参照画の各パーツの色を統一する処理を行った．また，線画は XDoG[20]を用いてアニメ画像から抽出したものを用いた．

さらに，今回は中間特徴量に VGG19 のものを用いた場合と Illustration2Vec[21]のものを用いた場合で比較する．

4.2 比較実験

比較実験の対象として，Maejima らの手法を選択した．比較実験においても本手法と同じ参照画 3 枚を用いて線画 7 枚に対して着色を行う．

4.3 評価

評価指標には，色ごとの Intersection over Union (IoU)のスコアを平均して求められる mean IoU (mIoU)を用いる．mIoU を用いる理由として，色の使用される頻度がそれぞれ異なるということが挙げられる．使われる頻度の少ない色が存在する場合でも mIoU は有効に評価に用いることが出来る．

mIoU は次の式によって求めることが出来る．

$$\mathrm{mIoU} = \frac{1}{K}\sum_{i=1}^{K}\frac{N_{ii}}{\sum_{i=1}^{K}\left(N_{ij}+N_{ji}\right)-N_{ii}}$$

Kは画像内に存在する色の数，N_{ij}は正解の色がiである場合に出力画像においてjの色で塗られている画素の数を表す．

Vol.2021-CG-184 No.16
Vol.2021-DCC-29 No.16
Vol.2021-CVIM-227 No.16
2021/11/5

mIoU が 1 に近いほど，正解画像に近い画像を出力できているということになる．

5. 実験結果

	比較対象	VGG19 を用いた場合	Illustration2Vec を用いた場合
mIoU の平均	0.464	0.822	0.856

表 1　実験結果

実験結果を表 1 に示す．表 1 においては比較対象である Maejima らの手法の場合，提案手法を VGG19 の重みを用いて行った場合, 提案手法を Illustration2Vec の重みを用いて行った場合のそれぞれにおいて，mIoU の平均を示している．また，自動彩色の結果の例を図 4 に示す．

5.1 比較対象と提案手法の比較

mIoU の平均を算出した結果，Maejima らの手法より提案手法を用いた場合の方が性能が良いことが分かった．Maejima らの手法はグラフマッチングをベースとしているため，次にあげられるような弱点が存在する．

まず，腕が動くなどして領域の数が参照画と比べて大きく変化すると領域の対応付けがうまくいかないことが挙げられる．次に，領域の対応付けが一つうまくいかないと，連鎖的に他の領域の対応付けを間違ってしまうということが挙げられる．後者の弱点は特に図 3 の右下の例において顕著に表れたため，mIoU が極端に低く表れている．

一方，提案手法においては画素ごとに独立して参照画像との対応付けを行っているため，ある画素の対応付けがうまくいかなくても別の画素との対応付けに影響することがない．そのため，mIoU が極端に低くなるような彩色結果は現れなかった．

5.2 VGG19 の重みを使用した場合と Illustration2Vec の重みを使用した場合の比較

mIoU の平均を算出した結果，提案手法で Illustartion2Vec の重みを使用した場合の方が VGG19 の重みを使用した場合よりわずかに性能が良いことが示された．理由としては，線画は実画像よりもイラスト寄りの性質を持った画像であることが挙げられる．VGG19 は猫や犬などを分類するために大量の実画像を訓練画像として学習を行ったモデルである．それに対して，Illustration2Vec は画像に映っているキャラクターがツインテールであるかどうかなどのイラスト画像の特徴を判別するために大量のイラスト画像を訓練画像として学習を行ったモデルである．線画は黒の線と白い領域のみで構成されており，実画像よりもイラスト画像に近いため，VGG19 の重みを使用するよりも Illustration2Vec の重みを使用する方が適切な中間特徴量を出力できると考えられる．そのため，参照画の線画と着色対象の線画との対応関係は，VGG19 の重みを使用するよりも Illustration2Vec の重みを使用する方適切であると考えられる．よって，VGG19 の重みを使用するよりも Illustration2Vec の重みを使用する方が mIoU の平均が高くなったと考えられる．

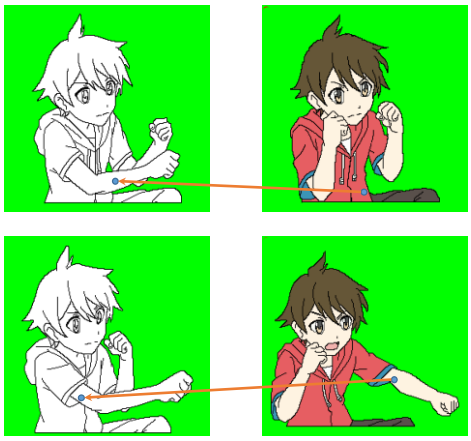

図 5　対応関係が不安定なケース

また，先述の議論を拡張するならば，線画を訓練画像として学習を行ったモデルは線画を入力しとしたときにより適切な中間特徴量を出力できると考えられる．そのため，線画を訓練画像として学習を行ったモデルを用いて提案手法を行った場合に更なる精度向上が期待できると考えられる．

6. 課題

6.1 対応関係が不安定なケース

図 5 に対応関係が不安定であったケースを示す．

図 5 の上の行においては，想定されている箇所とは全く異なる箇所から色を取ってきてしまっている．これは，参照画の線画と着色対象の線画が大きく異なることによって，対応付けがうまくいかなかったと考えられる．

図 5 の下の行においては，着色対象の線画の左腕と対応している箇所が参照画の右腕となっている．これは，線画の形が左腕と右腕で似ているために起きてしまっていると考えられる．今回は左腕と右腕の色が同じキャラクターであったため結果的に着色に影響が出ることはなかったが，左腕と右腕の色が異なるキャラクターにおいては影響が出ると考えられる．

6.2 着色時間

提案手法においては参照画が 3 枚の場合，対応付けの時間が 1 枚につき 20 分以上かかってしまっている．着色時間を減らすために以下のような方法が考えられる．

①対応関係の密度を少なくする

現在は対応関係を着色対象の線画の全てのピクセルにおいて取っている．この対応関係を取るピクセルの数を減ら

すことによって，対応付けの時間を減らすことが出来ると考えられる．しかし，対応付けの密度を減らすと面積の小さい領域における着色の精度が下がる可能性があるため，対応付けの時間と着色の精度のトレードオフになると考えられる．

②キャラクターが映っていない部分の対応付けを行わない

着色対象の線画においてキャラクターが映っていない部分の対応付けは意味がないため，キャラクターが映っていない部分の対応付けを省略することで対応付けの時間短縮が見込める．この方法が画像中に占めるキャラクターの割合が少なければ少ないほど機能すると考えられる．一方で，画像中に占めるキャラクターの割合が多いと，キャラクターが映っているか映っていないかの判定の時間によって対応付けにかかる時間が長くなると考えられる．

7. まとめ

本論文では，画素対応を用いることによって自動着色を行った．複数の参照画の線画と着色対象の線画との画素対応を取り，その対応関係の中で最も確からしい対応関係を選んで，その対応関係から画素の色を決定するという手法である．

従来手法と比較すると mIoU の観点から性能が良いことが示された．一方で，参照画の線画と着色対象の線画が大きく異なると対応付けが難しかったり，対応付けにかかる時間が長かったりするという弱点も存在した．参照画の線画と着色対象の線画がどの程度異なるとうまく対応付けが出来なくなるのか，また異なる度合いを定量化できないかどうかを調査していきたいと思う．

8. 謝辞

本研究は，JST 未来社会創造事業(JPMJMI19B2)および JSPS 科研費(JP19H01129)の支援を受けています。

参考文献

[1] Zhang. L, Li. C, Wong. T, Ji. Y, Liu. C, “Two-stage sketch colorization”, SIGGRAPH Asia 2018.
[2] Zhang. L, Li. C, Simo-Serra, E, Ji. Y, Wong. T, Liu. C, “User-Guided Line Art Flat Filling with Split Filling Mechanism”, CVPR 2021.
[3] Shi. M, Zhang. J, Chen. S, Gao. L, Lai. Y, Zhang. F, “Deep line art video colorization with a few references”, 2020.
[4] Zhang. Q, Wang. B, Wen. W, Li. H, Liu. J, “Line Art Correlation Matching Feature Transfer Network for Automatic Animation Colorization”, 2020.
[5] 前島謙宣, 久保尋之, 品川政太朗, 船冨卓哉, 四倉達夫, 中村哲, 向側康博, アニメに特化したサンプリングによるパッチベース学習に基づく着色作業支援, VC2021.
[6] Liu.Y, Qin. Z, Luo. Z, Wang. H, “Auto-painter: Cartoon Image Generation from Sketch by Using Conditional Generative Adversarial Networks”, 2017.
[7] 品川政太朗, 久保尋之, 石井大地, 前島謙宣, 船富卓哉, 中村哲, 向側康博, セマンティックセグメンテーションに基づくアニメ作品の自動彩色, VC2020.
[8] Sato. K, Matsui. Y, Yamasaki. T, Aizawa K, “Reference-based manga colorization by graph correspondence using quadratic programming”, SIGGRAPH Asia 2014.
[9] Maejima. A, Kubo. H, Funatomi. T, Yotsukura. T, Nakamura. S, Mukaigawa. Y, “Graph matching based anime colorization with multiple references”, SIGGRAPH 2019.
[10] Sykora. D, Ben-chen. M, Cadik. M, Whited. B, Simmons. M, “TexToons: practical texture mapping for hand-drawn cartoon animations”, NPAR 2004.
[11] Liu. C, Yuen. J and Torralba. A, “Sift flow: Dense correspondence across scenes and its spplications”, IEEE Transaction.
[12] Barnes. C, Shechtman. E, Finkelstein. A and Goldman.D.B, “PatchMatch: A randomized correspondence algorithm for structural image editing”, In proc of SIGGRAPH, 2009.
[13] Lowe. G. E, “Distinctive image features from scale-invariant keypoints”, In proc of IJCV, 2004.
[14] Liao. J, et al., “Visual attribute transfer through deep image analogy”, In proc of SIGGRAPH, 2017.
[15] Aberman. K, et al., “Neural best-buddies: sparse cross-domain correspondence”, In proc of SIGGRAPH, 2018.
[16] Simonyan. K and Zisserman A, “Very deep convolutional networks for large scale image recognition”, In proc of ICLR, 2015.
[17] 沖川翔太, 山口周悟, 森島繁生, アニメ制作過程における画素対応を用いた作画ミス検出, 第 83 回情報処理学会全国大会, 2021.
[18] Yamaguchi. Y and Morishima S, “Face texture synthesis from multiple images via sparse and dense correspondence”, In proc of SIGGRAPH Asia, 2016.
[19] Zhang. S, Chen. T, Zhang.Y, Hu.S, and Martin. R. R, “Vectorizing cartoon animations,” IEEE Transactions on Visualization and Computer Graphics,vol.15,no.4,pp. 618–629, 2009.
[20] Winnemoller. H, et al, “XDoG: an extended difference-of-gaussians compendium including advanced image stylization”, 2012.
[21] Saito. M and Matsui. Y, “Illustration2-Vec: a semantic vector representation of Illustrations”, In proc of SIGGRAPH Asia, 2015.